D0 Upgrade Electronics

New FE L1 Trigger CFT Design Version 1

by Fred Borcherding

Revised CFT L1 track finder algorithm

A revised CFT track finder algorithm has been developed. The former algorithm, which is part of the base line design, reported out a list of six tracks. For each of these tracks an index is given which represents which angular, phi, and momentum, Pt, bin the track belonged to. The size or resolution of this bin in delta phi and delta Pt was satisfactory for the needs of the muon L1 trigger and for a stand alone CFT L1 (or L2) trigger. But for the SVTpp and for matching with other sub-detectors in the L2 global this resolution was not precise enough.

The base line design algorithm found tracks in two dimensional, 2d, bins of 16 bins in phi by 4 bins in Pt. A Pt and phi index was assigned each of these 64 bins and the 6 highest Pt tracks from this list of from zero to 64 tracks was loaded into a six deep track FIFO for transmission to the rest of the trigger system. To assign an index to each of these 64 possible tracks the list of 64 was divided into 8 groups of 8. The 8 bits of each group was used as an address to enter a ROM. Each memory address of this ROM contained the appropriate bits for a six deep FIFO of track indexes corresponding to the tracks found for the bit pattern of the address. This method is very fast but each ROM requires a great deal of resources. Expanding the 2d track bins beyond 64 is impossible and a different methodology was sought.

The advantage of the ROM solution is that it in effect does a sort for six items over 64 bins in parallel. 64 bins can be scanned within a couple of clock cycles instead of six times 64 clock cycles. An alternative to this sort is to expand the number of bins so that the probability of more than one track in each bin is small. Then the bins are arranged within the code in the order in which the final track list is to be sorted. Therefore no sorting is necessary. That is what the revised algorithm attempts to do.

Fred Borcherding 1

| | Pt Index | | | | | | | | | | |
|-----------|----------|-----|-----|-----|-----|-----|-----|-----|-----|------|------|
| Phi Index | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 23 | 24 |
| 1 | 101 | 201 | 301 | 401 | 501 | 601 | 701 | 801 | 901 | 2301 | 2401 |
| 2 | 102 | 202 | 302 | 402 | 502 | 602 | 702 | 802 | 902 | 2302 | 2402 |
| 3 | 103 | 203 | 303 | 403 | 503 | 603 | 703 | 803 | 903 | 2303 | 2403 |
| 4 | 104 | 204 | 304 | 404 | 504 | 604 | 704 | 804 | 904 | 2304 | 2404 |
| 5 | 105 | 205 | 305 | 405 | 505 | 605 | 705 | 805 | 905 | 2305 | 2405 |
| 6 | 106 | 206 | 306 | 406 | 506 | 606 | 706 | 806 | 906 | 2306 | 2406 |
| 7 | 107 | 207 | 307 | 407 | 507 | 607 | 707 | 807 | 907 | 2307 | 2407 |
| 8 | 108 | 208 | 308 | 408 | 508 | 608 | 708 | 808 | 908 | 2308 | 2408 |
| 9 | 109 | 209 | 309 | 409 | 509 | 609 | 709 | 809 | 909 | 2309 | 2409 |
| 10 | 110 | 210 | 310 | 410 | 510 | 610 | 710 | 810 | 910 | 2310 | 2410 |
| 11 | 111 | 211 | 311 | 411 | 511 | 611 | 711 | 811 | 911 | 2311 | 2411 |
| 12 | 112 | 212 | 312 | 412 | 512 | 612 | 712 | 812 | 912 | 2312 | 2412 |
| 13 | 113 | 213 | 313 | 413 | 513 | 613 | 713 | 813 | 913 | 2313 | 2413 |
| 14 | 114 | 214 | 314 | 414 | 514 | 614 | 714 | 814 | 914 | 2314 | 2414 |
| 15 | 115 | 215 | 315 | 415 | 515 | 615 | 715 | 815 | 915 | 2315 | 2415 |
| 16 | 116 | 216 | 316 | 416 | 516 | 616 | 716 | 816 | 916 | 2316 | 2416 |
| 17 | 117 | 217 | 317 | 417 | 517 | 617 | 717 | 817 | 917 | 2317 | 2417 |
| | | | | | | | | | | | |
| 47 | 147 | 247 | 347 | 447 | 547 | 647 | 747 | 847 | 947 | 2347 | 2447 |
| 48 | 148 | 248 | 348 | 448 | 548 | 648 | 748 | 848 | 948 | 2348 | 2448 |

Table 1

The equations are sorted by Pt value into 24 bins and by Phi index into 48 bins giving 1152 2d bins. Each 2d bin is assigned an index number that codes its phi bin value and its Pt bin value. Table 1 above shows the bins and an example of how the index is assigned. (In this example the phi and Pt indexes are coded as two digit decimal numbers. In the PLD they are coded as bits.) The sorting criterion is arbitrary. The equations could be sorted for example according to the mean Pt of the equations and the mean phi value at a particular radius. They could be sorted by outer fiber bin and inner fiber bin. They could be sorted by outer fiber bin and offset number. All that is required is that 2d bins of 48 phi units by 24 Pt units are formed.

If any equation in a 2d bin is true, that is a track is found for any equation in that group, the 2d bin is set as true. Next the bins are scanned in groups of 6 over the Pt index, low index to high index. One scan group of the total 192 is shaded in gray in table 1. The scan is made from low Pt index to high until a 2d bin which is true is found. When a true is found the scan is stopped and the index of that first 2d bin is recorded. Any other tracks which may be in this group are lost.

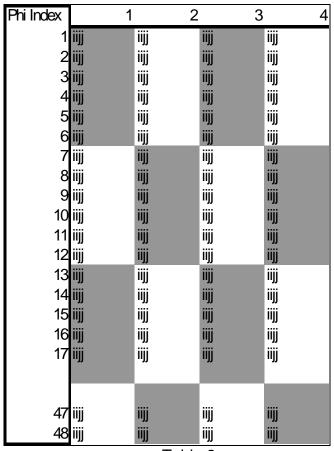


Table 2.

Table 2 above shows the result of this first step. The 2d bin index for each of the groups of 6 2d bins is recorded as index iijj. If there was one or more found track in any of the table entries above the value of iijj is the 2d bin tag for the lowest Pt index track. If there was no track the value of iijj is zero.

Next groups of 6 indexes are zero suppressed and loaded into FIFO's. Examples of these groups are shown as shaded in table 2 above. After this step there are 32 FIFO's each with from zero to six track indexes.

| | Pt Index | | | | |
|-----------|----------|------|------|------|---|
| Phi Index | 1 | 2 |) | 3 | 4 |
| 1 | fifo | fifo | fifo | fifo | |
| 2 | fifo | fifo | fifo | fifo | |
| 3 | fifo | fifo | fifo | fifo | |
| 4 | fifo | fifo | fifo | fifo | |
| 5 | fifo | fifo | fifo | fifo | |
| 6 | fifo | fifo | fifo | fifo | |
| 7 | fifo | fifo | fifo | fifo | |
| 8 | fifo | fifo | fifo | fifo | |

Table 3

Table 3 above shows these 32 FIFO's. Remember that each of these FIFO's is from zero to six tracks long. Next the FIFO's are merged in pairs. The shaded areas in the above table illustrate this. All the tracks from the first FIFO are loaded into a new FIFO. Any open slots in the new FIFO are then loaded with tracks from the second FIFO. This process is from low index to high index value. At the end of this step there are 16 FIFO's.

| | Pt Index | | | | | |
|-----------|----------|------|------|------|--|--|
| Phi Index | 1 | 2 | 3 | 3 4 | | |
| 1 | fifo | fifo | fifo | fifo | | |
| 2 | fifo | fifo | fifo | fifo | | |
| 3 | fifo | fifo | fifo | fifo | | |
| 4 | fifo | fifo | fifo | fifo | | |

Table 4 shows these 16 FIFO's. The same process is repeated again for these 16 FIFO's as shown by the example shaded area in table 4. At the end of this step there are 8 FIFO's.

| | Pt Index | | | | |
|-----------|----------|------|------|------|--|
| Phi Index | 1 | | 2 | 3 | |
| | fifo | fifo | fifo | fifo | |
| 2 | fifo | fifo | fifo | fifo | |

Table 5 above shows these 8 FIFO's, which are in turn load in pairs.

| | Pt Index | | | | | | | |
|-----------|----------|------|------|------|---|--|--|--|
| Phi Index | | 1 | 2 | 3 | 4 | | | |
| 1 | fifo | fifo | fifo | fifo | | | | |

Table 6 shows the result of this combining. At this stage the FIFO's are completely collapsed in phi and we have four FIFO's each of which is for a separate Pt range. These four FIFO's are combined in pairs down to two.

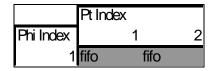


Table 7 shows these two FIFO's. They are in turn combined into a single FIFO.

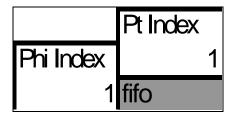


Table 8 shows the final, single 6 track deep FIFO. And the process is finished.

Realization of the Algorithm in PLD

A realization of the algorithm from the 24 by 48 array of binary bits to the final single 6 deep track FIFO has been done. Xxxxxx......

| | Pt Index | | | | | | | | | | |
|-----------|----------|-----|-----|-----|-----|-----|-----|-----|-----|------|------|
| Phi Index | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 23 | 24 |
| 1 | 101 | 201 | 301 | 401 | 501 | 601 | 701 | 801 | 901 | 2301 | 2401 |
| 2 | 102 | 202 | 302 | 402 | 502 | 602 | 702 | 802 | 902 | 2302 | 2402 |
| 3 | 103 | 203 | 303 | 403 | 503 | 603 | 703 | 803 | 903 | 2303 | 2403 |
| 4 | 104 | 204 | 304 | 404 | 504 | 604 | 704 | 804 | 904 | 2304 | 2404 |
| 5 | 105 | 205 | 305 | 405 | 505 | 605 | 705 | 805 | 905 | 2305 | 2405 |
| 6 | 106 | 206 | 306 | 406 | 506 | 606 | 706 | 806 | 906 | 2306 | 2406 |
| 7 | 107 | 207 | 307 | 407 | 507 | 607 | 707 | 807 | 907 | 2307 | 2407 |
| 8 | 108 | 208 | 308 | 408 | 508 | 608 | 708 | 808 | 908 | 2308 | 2408 |
| 9 | 109 | 209 | 309 | 409 | 509 | 609 | 709 | 809 | 909 | 2309 | 2409 |
| 10 | 110 | 210 | 310 | 410 | 510 | 610 | 710 | 810 | 910 | 2310 | 2410 |
| 11 | 111 | 211 | 311 | 411 | 511 | 611 | 711 | 811 | 911 | 2311 | 2411 |
| 12 | 112 | 212 | 312 | 412 | 512 | 612 | 712 | 812 | 912 | 2312 | 2412 |
| 13 | 113 | 213 | 313 | 413 | 513 | 613 | 713 | 813 | 913 | 2313 | 2413 |
| 14 | 114 | 214 | 314 | 414 | 514 | 614 | 714 | 814 | 914 | 2314 | 2414 |
| 15 | 115 | 215 | 315 | 415 | 515 | 615 | 715 | 815 | 915 | 2315 | 2415 |
| 16 | 116 | 216 | 316 | 416 | 516 | 616 | 716 | 816 | 916 | 2316 | 2416 |
| 17 | 117 | 217 | 317 | 417 | 517 | 617 | 717 | 817 | 917 | 2317 | 2417 |
| | | | | | | | | | | | |
| 47 | 147 | 247 | 347 | 447 | 547 | 647 | 747 | 847 | 947 | 2347 | 2447 |
| 48 | 148 | 248 | 348 | 448 | 548 | 648 | 748 | 848 | 948 | 2348 | 2448 |